COMP 3000 Operating Systems

Study Session Questions

Date: _____/ _____/ _____

Instructions:

- Answer all questions.
- Show all work for partial credit where appropriate.
- No electronic devices allowed.

Part 1 - Overview (OSC Ch. 1-2)

1. Which of the following statements are true? **Select all that apply.**
   A. An interrupt is ignored if another interrupt is currently being handled.
   B. It's possible for two devices to each generate interrupts at the same time.
   C. Division by zero always causes an interrupt.
   D. All interrupts are generated only by hardware devices.
   E. Disabling interrupts guarantees mutual exclusion on all systems.
   F. A system call is an example of a software interrupt.

2. Place the steps of interrupt handling in the correct order, from the beginning. (1, 2, 3, 4)
   ☐ the CPU uses the interrupt vector table to find the right interrupt handler
   ☐ the device controller/software sends the interrupt to CPU
   ☐ the CPU restores its state and resumes normal operation
   ☐ the CPU finishes its ongoing instruction and saves current program state

3. Place the following storage systems in order of relative typical speed, from slowest to fastest. (1, 2, 3, 4)
   ☐ Hard-disk drives
   ☐ Registers
   ☐ Cache
   ☐ Main memory

4. What is the main difference between an exception and an interrupt?

5. Consider the following list of actions. Which of the following should be performed by the kernel, and not by user programs? Select all that apply.
   A. reading the value of the program counter (PC).
   B. changing the value of the program counter (PC).
   C. increasing the size of an address space.
   D. creating a memory segment that is shared between multiple processes.
   E. writing to a memory segment that is shared between multiple processes.
   F. disabling interrupts.

6. Describe three primary functions of an operating system (what does it do?)

Part 2 - Process Management (OSC Ch. 3-5)

7. Which of the following statements are true? **Select all that apply.**
    A. It is possible to have concurrency **but not parallelism**.
    B. Threads within the same process share the same **address space**.
    C. The **Ready** queue contains processes that are waiting for I/O to complete.
    D. In preemptive priority scheduling, starvation of low-priority processes is **impossible**.
    E. A race condition occurs when the correctness of a program depends on the relative timing of threads.
    F. A successful context switch **always** requires saving the state of the currently running process and restoring the state of the next process.

8. What is the use of the return value of `fork()`?

9. Which of the following is shared between threads of the same process? Select all that apply.
    A. integer and floating point registers
    B. program counter
    C. heap memory
    D. stack memory
    E. global variables
    F. open files

10. Under which of the following basic CPU scheduling policies is starvation possible? Select all that apply.
    A. First-Come-First-Served (FCFS)
    B. Shortest-Job-First (SJF)
    C. Shortest-Remaining-Time-First (SRTF)
    D. Round-Robin (RR)

11. What is *context switching*?

12. What is the distinction between a ready thread and a waiting thread?

13. Consider the following set of processes, with the length of the CPU burst time given in milliseconds:

Process	Burst Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0.

Which of the following algorithms results in the minimum average waiting time (over all processes)?

A. First-Come-First-Served (FCFS)

B. Shortest-Job-First

C. Non-preemptive Priority (with a larger priority number implying a higher priority)

D. Round-Robin (quantum = 2)

Use the space below to justify your solution.

14. A system uses an SRT (Shortest Remaining Time) scheduler. Among processes with the shortest remaining time, the earliest-arriving one is chosen.

    Three processes arrive: P_0, P_1, and P_2. P_0 has a CPU burst of 4 units, P_1 has a burst of $(1+y)$ units, P_2 has a burst of 3 units.

    The processes arrive at times 0, 2, and 5 (one process at each time).

    (a) Determine the arrival order of P_0, P_1, and P_2.

    (b) Find the value of y so that:

    - SRT produces the **same** schedule as FCFS.
    - The average response time of the three processes is **maximized**.

    Response time is the time from arrival to first execution.

Part 3 - Process Syncronization (OSC Ch. 6-8)

15. Match each term to the most accurate and precise defenition.

A. Mutual Exclusion	**1.** Each process waits at most a finite number of turns.
B. Bounded Waiting	**2.** If the critical section is empty, waiting processes must eventually enter.
C. Progress	**3.** Only one process may be in its critical section.

16. (a) How could one PREVENT deadlock in the situation shown below using deadlock prevention techniques discussed in class?

Four buses blocking each other at the Alexander Klellands Plass intersection in Oslo, Norway. Photo dated 24 November 2025.

(b) Provide another (non-operating-system-related) example of deadlock.

17. Consider the following information about resources in a system:

- There are two classes of allocatable resource labelled R1 and R2.
- There are two instances of each resource.
- There are four processes labelled P1 through P4.
- There are some resource instances already allocated to processes, as follows:
  - one instance of R1 held by P2, another held by P3
  - one instance of R2 held by P1, another held by P4
- Some processes have requested additional resources, as follows:
  - P1 wants one instance of R1
  - P3 wants one instance of R2

(a) Draw the resource allocation graph for this system. Use the style of diagram from the lecture notes.

(b) What is the state (runnable, waiting) of each process? For each process that is waiting, indicate what it is waiting for.

(c) Is this system deadlocked? If so, state which processes are involved. If not, give an execution sequence that eventually ends, showing resource acquisition and release at each step.

18. Given the state below, determine whether the system is in a safe state. If it is safe, provide a safe sequence.

Allocation			
P	R_1	R_2	R_3
P0			
P1			
P2			
P3			
P4			

Max			
P	R_1	R_2	R_3
P0			
P1			
P2			
P3			
P4			

Need			
P	R_1	R_2	R_3
P0			
P1			
P2			
P3			
P4			

Available Vector			
	R_1	R_2	R_3
Available			

Part 4 - Memory Management (OSC Ch. 9-10)

19. Under what conditions might two or more processes share common physical frames? Explain how and why this would be done.

20. If every page in memory is accessed (used) between any two page faults, will the Clock replacement algorithm behave exatcly like FIFO? Justify your response whether true or false.

21. Assume that there are 5 pages, A, B, C, D, and E. Fill in the page reference string and complete the rest of the information in the table below so that LRU is the *worst* page replacement algorithm (i.e., it results in the maximum number of page faults). Use a dash "–" to fill in blank locations. Note that when there is more than one page that is a possible victim, always choose the one with the **lowest frame number**.

Num	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Refs	A	B	C	D	E										
Frame 1	A														
Frame 2	–														
Frame 3	–														
Frame 4	–														
Fault?	X														

22. Consider a virtual memory system that uses 2-level paging. The page size in this system is 256 (2^8) bytes. Each individual page table fits exactly into one memory frame, and the size of each page table entry (PTE) is 8 bytes.

    (a) What is the maximum size (in bytes) of a virtual address space in this system?

    (b) Suppose that there is a process with a virtual address space of the maximum size. How many bytes of memory are occupied by the page tables for this process?

    (c) Suppose that there is a process with a virtual address space of size 10240 ($10 \cdot 2^10$) bytes. Also suppose that the entire address space is in memory. How many valid PTEs will exist in the page tables for this process?

Part 5 - Storage Management (OSC Ch. 11-12)

Part 6 - File System (OSC Ch. 13-15)

Part 8 - Virtual Machines (OSC Ch. 18)